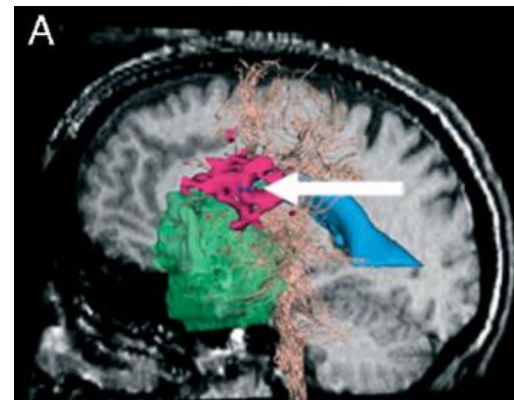


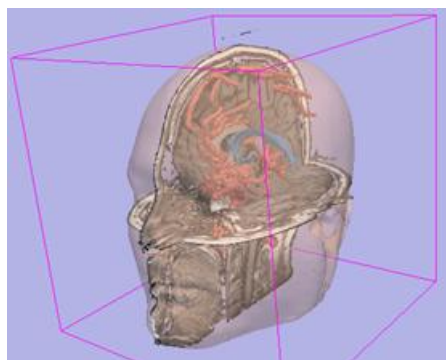
Może to?



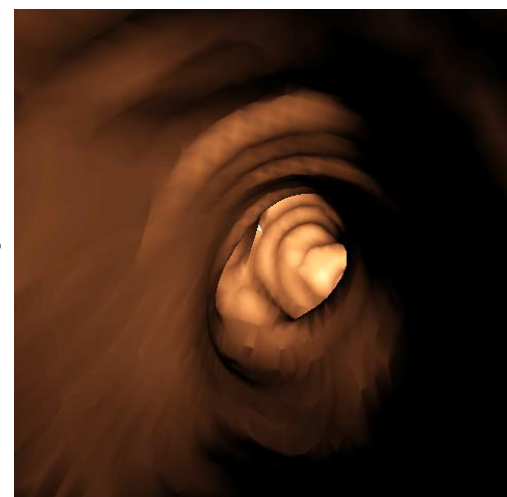
SZTUCZNA INTELIGENCJA

przygotował ze źródeł własnych i internetowych

Artur Przelaskowski



Może tam?

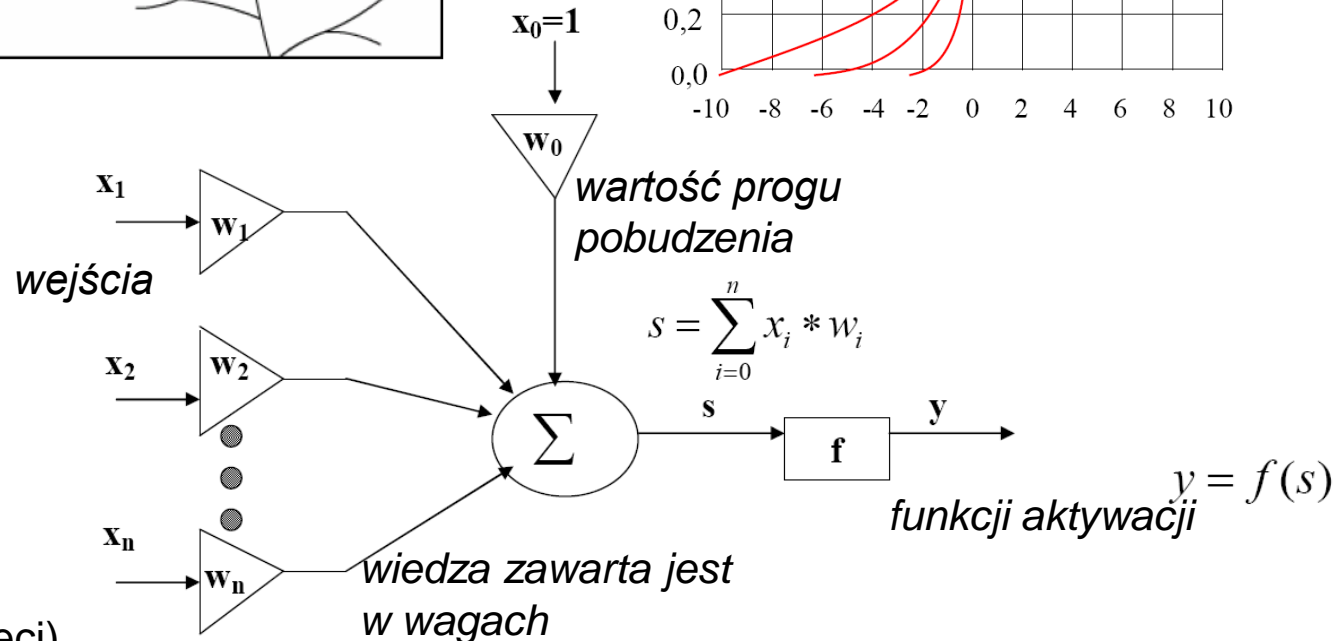
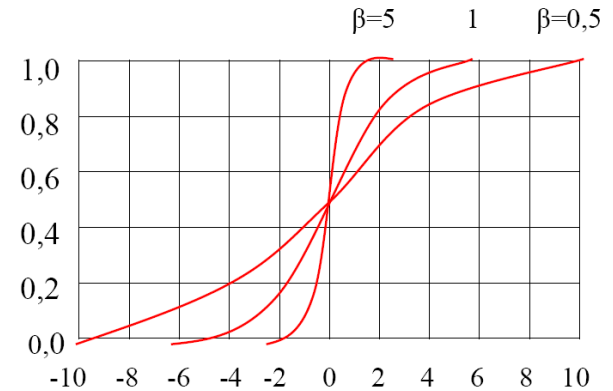
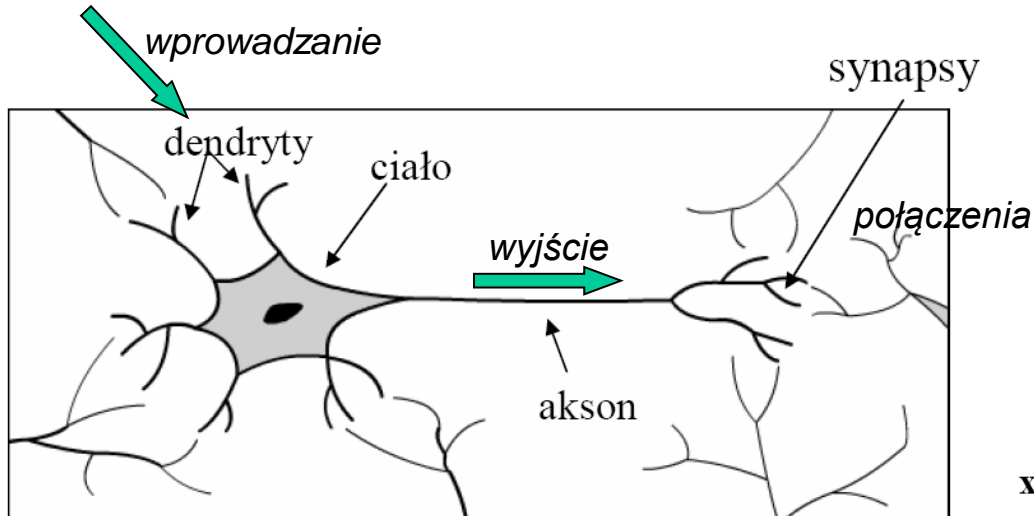


SIECI NEURONOWE

Pojęcie sieci neuronowych

- Obraz mózgu jako sieci neuronów realizującej równoległe, rozproszone, asynchroniczne przetwarzanie sygnałów
- Naśladowanie działania naturalnych neuronów
- Neuron przyjmuje wiele pobudzeń (dendryty), sumuje je i proguje (ciało komórki), a następnie wysyła (przez akson) do wszystkich podłączonych (synapsa) odbiorców (innych neuronów)
- Cechy:
 - Sieci neuronowe uczą się na przykładach (czasochłonny proces treningu, istotna rola reprezentatywności zbioru uczącego)
 - Przetwarzają dane w sposób równoległy
 - Potrafią poradzić sobie z zaszumionymi danymi
 - Są zdolne do generalizowania wiedzy szczegółowej, zdobytej w oparciu o przykłady
 - Struktury sieci są w dużym stopniu uniwersalne
 - Sieci działają bez reguł decyzyjnych, nie ma też reguł doboru struktury sieci do problemu (metoda prób i błędów)

Neuron i jego model (sztuczny)



Uczenie neuronu (dopasowanie struktury sieci) polega na modyfikacji wektorów wag w neuronach

Funkcje aktywacji

- Liniowe

- Nieliniowe nieciągłe

- Skok jednostkowy

$$f(s) = \begin{cases} 1 & \text{dla } s > 0 \\ 0 & \text{dla } s \leq 0 \end{cases}$$

- Skok jednostkowy bipolarny

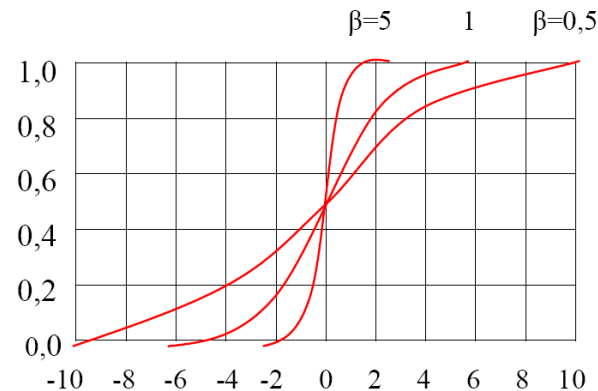
$$f(s) = \begin{cases} 1 & \text{dla } s > 0 \\ 0 & \text{dla } s = 0 \\ -1 & \text{dla } s < 0 \end{cases}$$

- Signum

- Nieliniowe ciągłe (wygodne w procesie optymalizacji)

- Sigmoidalna

$$f(s) = \frac{1}{1 + e^{-\beta * s}}$$



Perceptron (prosta funkcja aktywacji)

Rosenblatt 1958

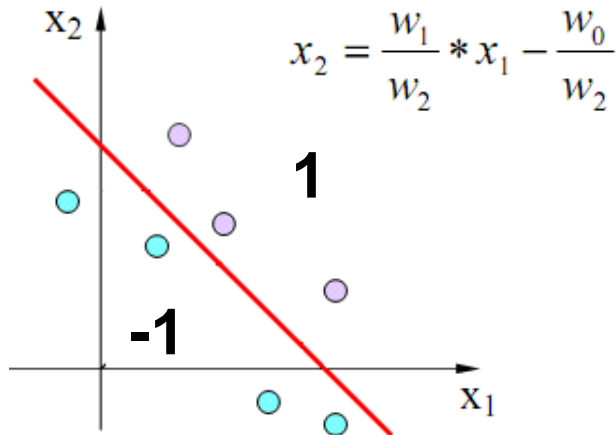
- Dwie klasy y
- Liniowa hiperpłaszczyzna rozdzielająca

$$\sum_{i=1}^n x_i * w_i + w_0 = 0$$

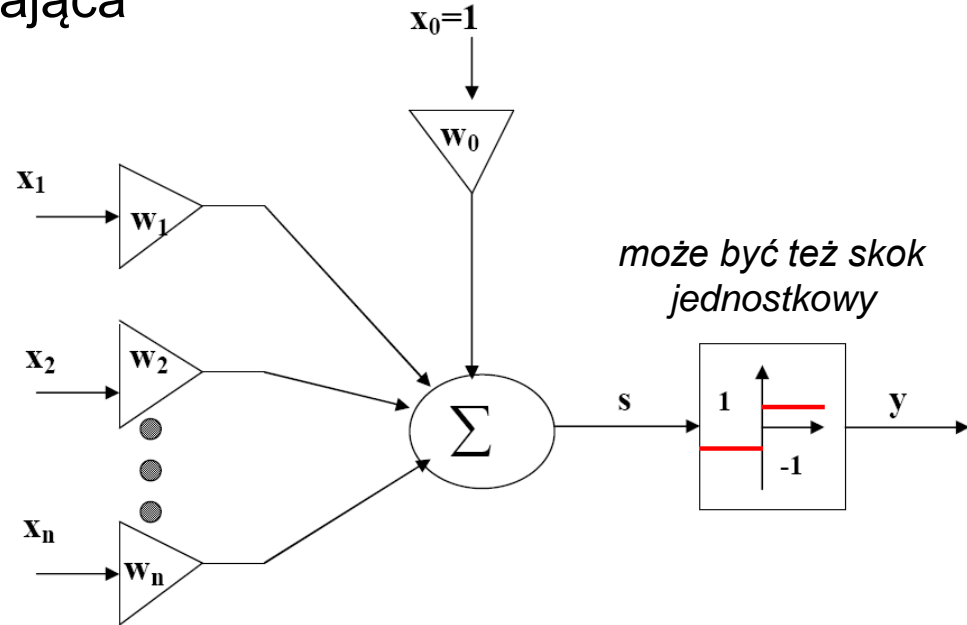
- Prosty, 2W przypadek

$$w_1 * x_1 + w_2 * x_2 + w_0 = 0$$

prosta podziału (granica decyzyjna)

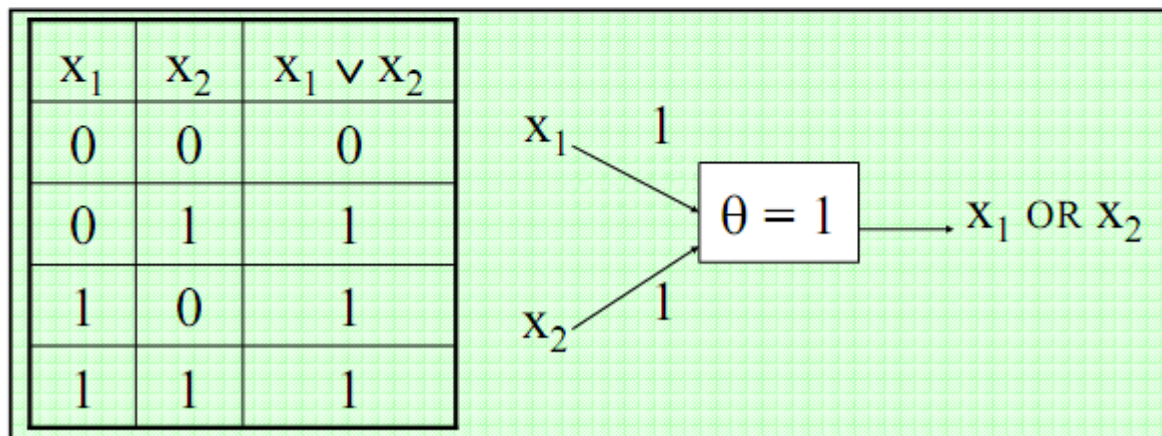
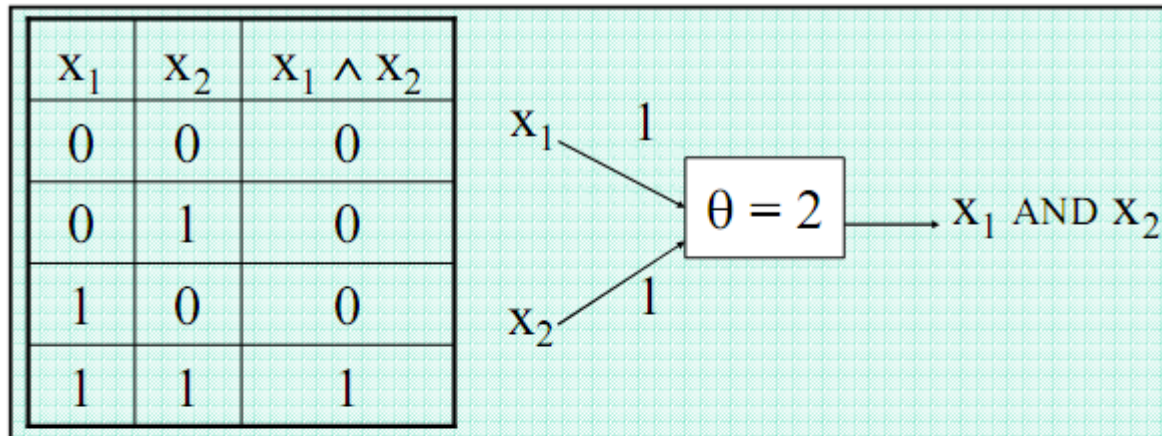


liniowa separowalność



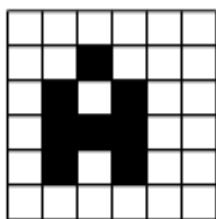
Rozwiązywanie jedynie prostych problemów klasyfikacji

Proste działanie perceptronu

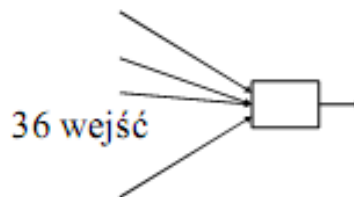


- Problem z XOR, które jest nieseparowalne liniowo
- Dopiero w połączeniu z innymi neuronami uzyskuje się zdolność podejmowania bardziej złożonych decyzji

Przykład – rozpoznawanie znaków



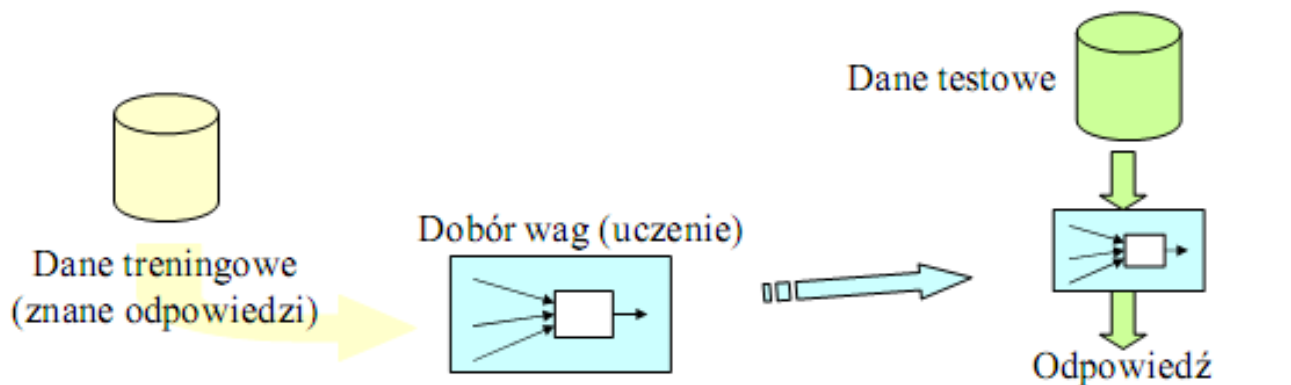
Siatka 6×6



36 wejść

Wyjście: 1, jeśli na wejściu pojawia się litera "A", zaś 0 w p.p.

Zadanie: dobrać wagi wejść i wartość progową tak, by uzyskać zaplanowany efekt



Uczenie perceptronu

- **Wejście:**

- Ciąg przykładów uczących ze znanymi odpowiedziami

- **Proces uczenia:**

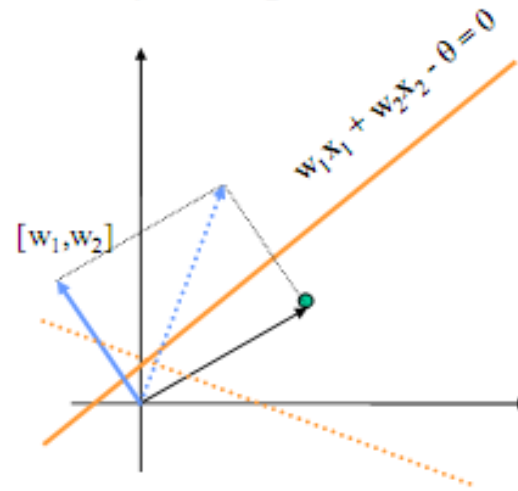
- Inicjujemy wagi losowo
- Dla każdego przykładu, jeśli odpowiedź jest nieprawidłowa, to

$$w_1 + = \alpha x_1$$

$$w_2 + = \alpha x_2$$

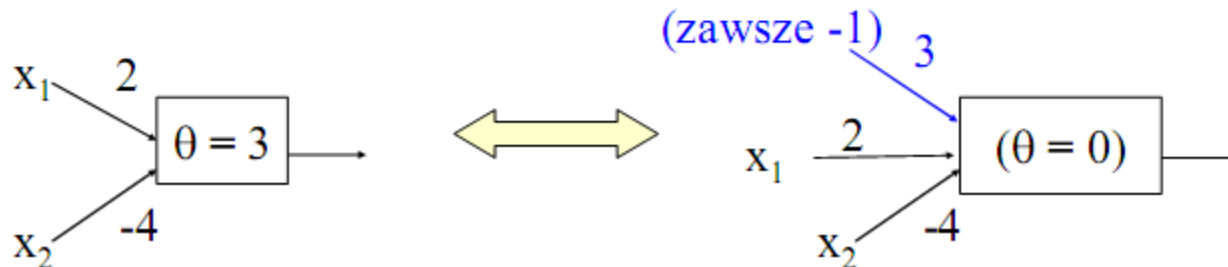
$$\theta - = \alpha$$

gdzie α jest równe różnicy między odpowiedzią prawidłową a otrzymaną.

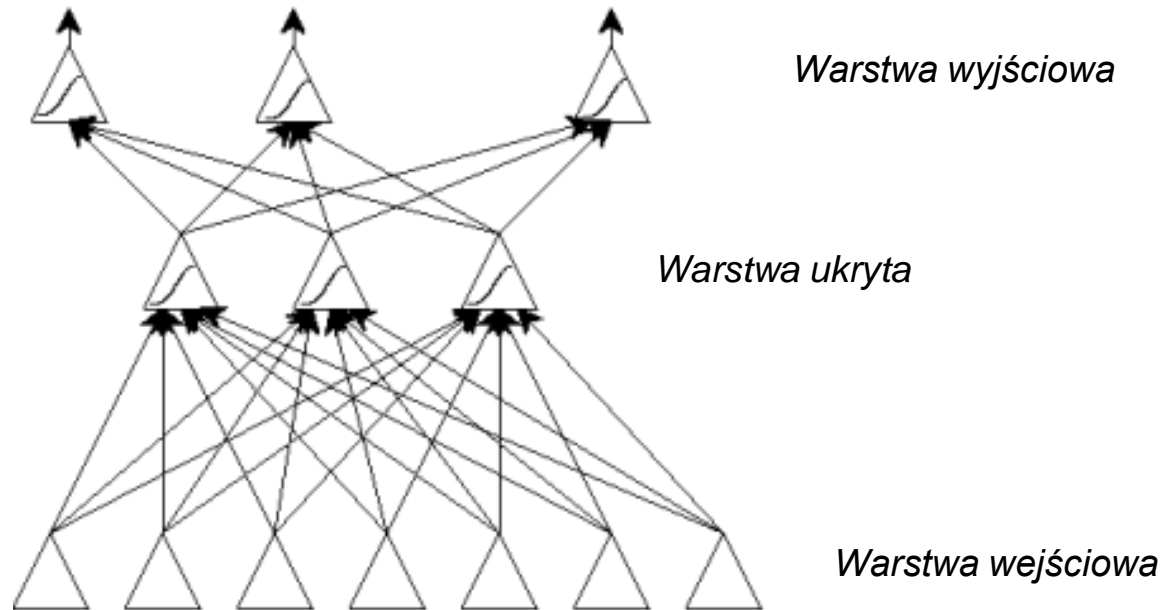


Uczenie perceptronu (2)

- Często α mnoży się dodatkowo przez niewielki współczynnik uczenia
- Po wyczerpaniu przykładów, zaczynamy proces uczenia od początku, dopóki następują jakiegokolwiek zmiany wag połączeń
- Próg θ można traktować jako wagę dodatkowego wejścia o wartości -1:



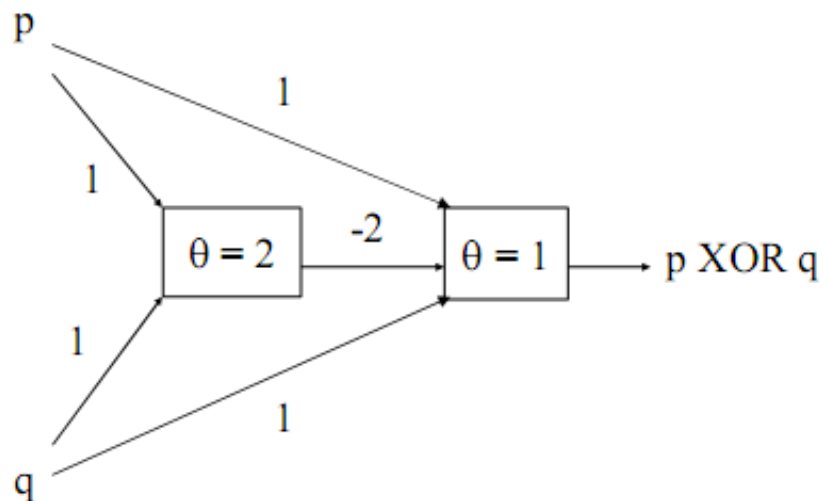
Sieć neuronowa (sieć perceptronów)



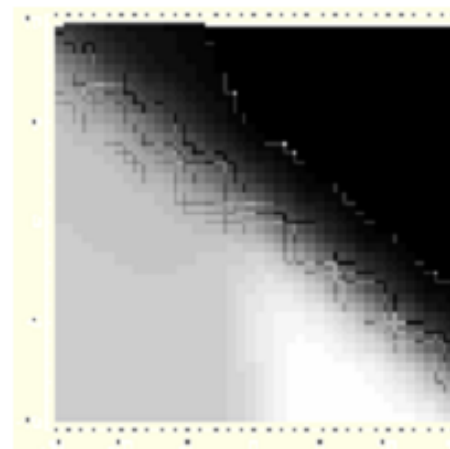
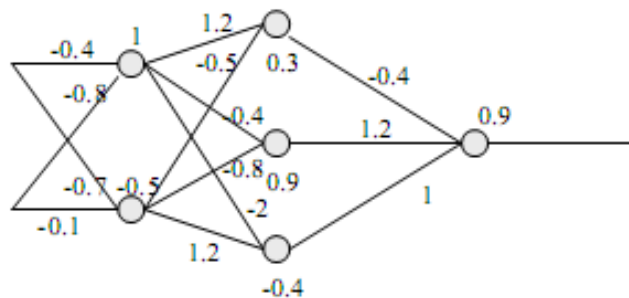
- działanie sieci polega na liczeniu odpowiedzi neuronów w kolejnych warstwach
- jeśli wszystkie poszczególne funkcje aktywacji są liniowe, to funkcja sieci jest również liniowa
- architektura wielowarstwowa daje zatem nowe możliwości tylko w przypadku stosowania funkcji nieliniowych

Teraz już można

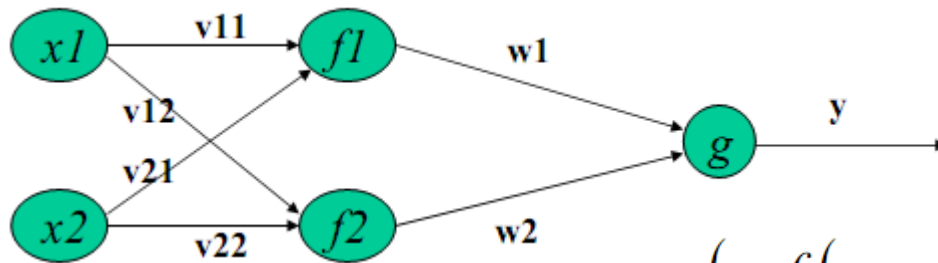
- XOR



- rozpoznawanie

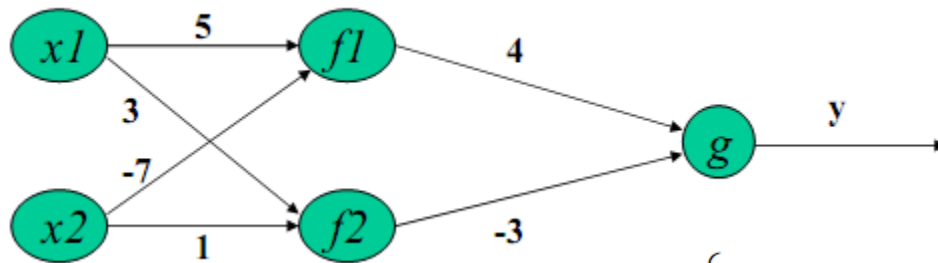


Sieci jako funkcje złożone



$$y = g(w_1 f_1(v_{11}x_1 + v_{21}x_2) + w_2 f_2(v_{12}x_1 + v_{22}x_2))$$

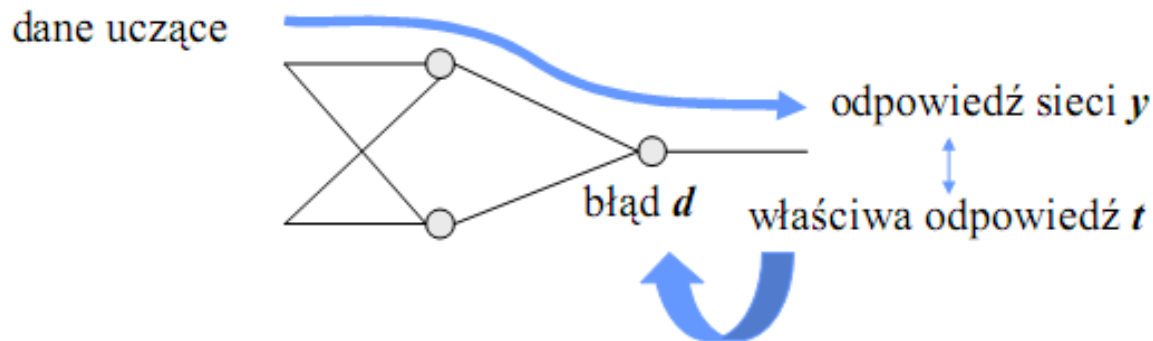
$$y = \text{Network}(x_1, x_2)$$



$$y = \begin{cases} 1 & \Leftrightarrow \frac{4}{1+e^{-3(5x_1-7x_2)}} - 3\left(\frac{2}{1+e^{-2(3x_1+x_2)}} - 1\right) \geq 8 \\ 0 & \Leftrightarrow \frac{4}{1+e^{-3(5x_1-7x_2)}} - 3\left(\frac{2}{1+e^{-2(3x_1+x_2)}} - 1\right) < 8 \end{cases}$$

Metoda propagacji wstecznej

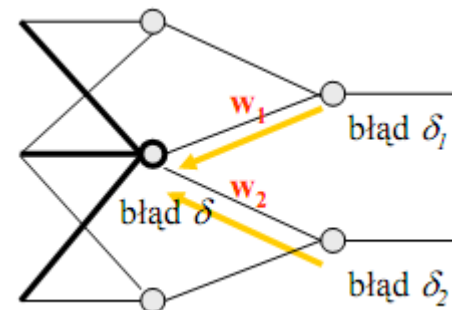
- Inicjujemy wagi losowo (na małe wartości)
- Dla danego wektora uczącego obliczamy odpowiedź sieci (warstwa po warstwie)
- Każdy neuron wyjściowy oblicza swój błąd, odnoszący się do różnicy pomiędzy obliczoną odpowiedzią y oraz poprawną odpowiedzią t



- typowa miara błędu

$$d = \frac{1}{2}(y - t)^2$$

$$d(w_1, \dots, w_K) = \frac{1}{2}(f(w_1 z_1 + \dots + w_K z_K) - t)^2$$



$$\Delta w = \eta \cdot \delta \cdot z$$